



## **PSP [20ES104] COURSE PROJECT REPORT**

On

### **PHONE BOOK**

Developed By:

**H.T.NO**

**STUDENT NAME**

**2203A51580**

**PEEKA SIDDARTHA**

Under the Guidance of

Dr. Mohammed Ali Shaik

Assistant Professor

Submitted to

Department Computer Science and Artificial Intelligence

SR University

Ananthasagar(V), Hasanparthy(M), Hanamkonda(Dist.) – 506371

[www.sru.edu.in](http://www.sru.edu.in)

**June 2023**

## Department of Computer Science and Artificial Intelligence

### CERTIFICATE

This is to certify that the PSP course project report entitled " **PHONEBOOK** " is a record of bonafide work carried out by the student **PEEKA SIDDARTHA** bearing roll number **2203A51580** of Computer Science and Artificial Intelligence department during the academic year **2022-23**.

#### **Supervisor**

(DR. MOHAMMED ALI SHAIK)

## INDEX

<b>Sl. No</b>	<b>Title</b>	<b>Page No.</b>
1.	Problem statement	4
2.	Module-wise description	7
3.	Knowledge required to develop the project	9
4.	Source code (.c file code followed by .h file code)	10
5.	Results	24

## PROBLEM STATEMENT:

Develop a C Application to Create a PHONE BOOK

Provide the functionality for below mentioned:

1. Read the "choice" of the **PHONEBOOK** details dynamically.
- 2.Sort the details accordingly to the function that are allocated

➤ displayMenu();

➤ addContact();

➤ searchContact();

➤ showAllContacts();

➤ deleteContact();

➤ editContact();

▷ saveContactsToFile();

▷ loadContactsFromFile();

▷ clearPhonebook();

### 3. Search "choice" of the diary details accordingly

▷ Add Contact

▷ Search Contacts

▷ Show All Contacts

▷ Delete a Contact

▷ Edit contact

▷ Save contacts to file

▷ Clear phonebook

▷ Exit

### 4. Print your "choice" of the diary details.

## ***MODULES:***

### **Add Contact:**

The program allows the user to add contacts dynamically by selecting the "Add Contact" option from the main menu.

The user is prompted to enter the contact's name, phone number, category and email.

Category and email are optional choice for the User.

The program does not include explicit sorting functionality for the phone book project. The contacts are displayed in the order they were added.

The user can add multiple contacts.

### **Search Contact:**

The program provides a "Search Contact" option in the main menu.

The user can enter a name to search for a specific contact.

If a match is found, the contact details (name, phone number, address, and email) are displayed.

The user can choose to edit the contact or go back to the main menu.

### **Show All Contacts:**

The program includes a "Show All Contacts" option in the main menu. When selected, all the contacts' details (name, phone number, address, and email) are displayed.

## **Delete a contact:**

The program allows the user to delete a contact by selecting the "Delete a Contact" option from the main menu.

The user is prompted to enter the name of the contact to be deleted.

If the contact exists, it is deleted from the phone book.

## **Edit a contact:**

The program provides the ability to edit a contact's details.

When searching for a contact, the user has the option to choose the "Edit Contact" option.

The user can select a field (name, phone number, address, or email) to edit and enter the new value.

The contact's details are updated accordingly.

## **Save contacts to file:**

The program provides the ability to Save contacts to file

This function opens a file named "phonebook.txt" in write mode ("w"), and if the file opening fails, it prints an error message and returns.

It then iterates over each contact in the all array and writes the contact's details to the file using `fprintf()`. Each contact's details are written on separate lines, and an empty line is added between contacts for better readability. Finally, the file is closed using `fclose()`.

You can call this function to save the contacts to the file whenever needed, such as when the user selects the "Save contacts to file" option in the menu.

## **Clear phonebook:**

The program provides the ability to clear phonebook. This function sets the total variable to 0, effectively clearing the phonebook and removing all contacts. It also displays a message to indicate that the phonebook has been cleared. User can call this function whenever needed, such as when the user selects the "Clear phonebook" option in the menu.

## **Exit the program:**

The user can choose the "Exit" option from the main menu to exit the program.



## **KNOWLEDGE REQUIRED TO DEVELOP THIS APPLICATION**

- Control Statements (if, if-else, switch)
- Loop Statements (while, for)
- Arrays (1D)
- Strings (Strings and Table of strings) and its functions  
(strcpy, strcmp)
- Functions (Any type of user defined functions)
- Structure (structures and nested structures)
- Pointers (pointer to strings and pointers to structures)
- Files

**SOURCE CODE [.C FILE]:**

```
#include<stdio.h>
#include<string.h>
#include<stdlib.h>

#define MAX_CONTACTS 250
FILE *ptr;

int i,n,total=6;
struct phonebook{
    char
    name[100],number[100],address[100],email[100],category[100],DOB[100];
    }a,all[MAX_CONTACTS];

int START();
int DELETE();
int EDIT_EMAIL();
int EDIT_NAME();
int EDIT_NUMBER();
int EDIT_ADDRESS();
int OPTIONS();
int ADD_CONTACT();
int SEARCH_CONTACT();
int ALL_CONTACTS();
int EDIT_CAT();
int EDIT_DoB();
int OPT();
int DELETE_ALL();
int DETAILS();
int SAVE_CONTACTS();

int main()
{
    DETAILS();
    START();
    SAVE_CONTACTS();
    return 0;
}

int START()
{
    while(1)
    {
```

```

    printf("\n-----WELCOME TO PHONEBOOK-----
---");
    printf("\nChoose");
    printf("\n1. Add contact");
    printf("\n2. Search contact");
    printf("\n3. Show all contacts");
    printf("\n4. Delete a contact");
    printf("\n5. Exit\n");
    scanf("%d",&n);
    switch (n)
    {
    case 1: ADD_CONTACT();
        break;
    case 2: SEARCH_CONTACT();
        break;
    case 3: ALL_CONTACTS();
        break;
    case 4: DELETE();
        break;
    case 5:
        printf("\nThank you!");
        return 0;
    default:printf("\nInvalid Input");
        break;
    }
}
return 0;
}

int DELETE()
{
    char temp1[20];
    int x;
    printf("\nEnter name : ");
    scanf("%s",temp1);
    printf("\nPress 1.Continue");
    printf("\nPress 2.Cancel\n");
    scanf("%d",&x);
    switch(x)
    {
    case 1:
        break;
    case 2:
        return 0;
    }
}

```

```
}
for(i=0;i<total;i++)
{
    if(strcmp(temp1,all[i].name)==0)
    {
        while(i<total)
        {
            if(i==total-1)
            {
                total--;
                break;
            }
            else
            {
                all[i]=all[i+1];
            }
            i++;
        }
        printf("\nContact successfully deleted");
        return 0;
    }
}
printf("\nContact not found");
return 0;
}

int EDIT_EMAIL()
{
    int x;
    printf("\nPresent email id : %s",all[i].email);
    printf("\nPress 1.Delete Email ID");
    printf("\nPress 2.Reset Email ID\n");
    scanf("%d",&x);

    switch(x)
    {
        case 1:
            strcpy(all[i].email,"__");
            return 0;
        case 2:
            break;
        default:
            printf("\nInvalid Input");
    }
}
```

```

    printf("\nEnter new email id : ");
    scanf("%s",all[i].email);
    printf("\nEmail id updated successfully");
    return 0;
}

int EDIT_NAME()
{
    printf("\nPresent name : %s",all[i].name);
    printf("\nEnter new name : ");
    scanf("%s",all[i].name);
    printf("\nName updated successfully");
    return 0;
}

int EDIT_NUMBER()
{
    printf("\nPresent number : %s",all[i].number);
    printf("\nEnter new number : ");
    scanf("%s",all[i].number);
    printf("\nNumber updated successfully");
    return 0;
}

int EDIT_ADDRESS()
{
    int x;
    printf("\nPresent address : %s",all[i].address);
    printf("\nPress 1.Delete Address");
    printf("\nPress 2.Reset Address\n");
    scanf("%d",&x);
    switch(x)
    {
        case 1:
            strcpy(all[i].address,"__");
            return 0;
        case 2:
            break;
        default:
            printf("\nInvalid Input");
    }
    printf("\nEnter new address : ");
    scanf("%s",all[i].address);
    printf("\nAddress updated successfully");
}

```

```
    return 0;
}

int EDIT_CAT()
{
    int select,x;
    printf("\nPresent category : %s",all[i].category);
    printf("\nPress 1.Delete Category");
    printf("\nPress 2.Reset Category\n");
    scanf("%d",&x);
    switch(x)
    {
        case 1:
            strcpy(all[i].email,"__");
            return 0;
        case 2:
            break;
        default:
            printf("\nInvalid Input");
    }
    printf("\nSelect category : ");
    printf("\n1.Family");
    printf("\n2.Friend");
    printf("\n3.Business");
    printf("\n4.Work");
    printf("\n5.Emergency");
    printf("\n6.Myself");
    printf("\n7.Other\n");
    scanf("%d",&select);
    switch(select)
    {
        case 1:strcpy(a.category,"Family");
            break;
        case 2:strcpy(a.category,"Friend");
            break;
        case 3:strcpy(a.category,"Business");
            break;
        case 4:strcpy(a.category,"Work");
            break;
        case 5:strcpy(a.category,"Emergency");
            break;
        case 6:strcpy(a.category,"Myself");
            break;
        case 7:strcpy(a.category,"Other");
```

```

        break;
        default:
        printf("\nInvalid Input");
        break;
    }
    printf("\nCategory updated successfully");
    return 0;
}

int EDIT_DoB()
{
    int x;
    printf("\nPresent DoB : %s",all[i].DOB);
    printf("\nPress 1.Delete DoB");
    printf("\nPress 2.Reset DoB\n");
    scanf("%d",&x);
    switch(x)
    {
        case 1:
            strcpy(all[i].DOB,"__");
            return 0;
        case 2:
            break;
        default:
            printf("\nInvalid Input");
    }
    printf("\nEnter DoB : ");
    scanf("%s",all[i].DOB);
    printf("\nDoB changed successfully");
}

int OPTIONS()
{
    int edit;
    if(all[i].number=="911")

    {
        printf("\nCan't edit this contact");
        return 0;
    }
    while(1)
    {
        printf("\nChoose");
        printf("\n1. Edit name");
        printf("\n2. Edit number");
    }
}

```

```
printf("\n3. Edit address");
printf("\n4. Edit email id");
printf("\n5. Edit category");
printf("\n6. Edit DoB");
printf("\n7. Delete");
printf("\n8. Exit\n");
scanf("%d",&edit);
switch(edit)
{
    case 1: EDIT_NAME();
        break;
    case 2: EDIT_NUMBER();
        break;
    case 3: EDIT_ADDRESS();
        break;
    case 4:
        EDIT_EMAIL();
        break;
    case 5:
        EDIT_CAT();
    case 6:
        EDIT_DoB();
    case 7:
        while(i<total){
            if(i==total-1)
            {
                total--;
                break;
            }
            else
            {
                all[i]=all[i+1];
            }
            i++;
        }
        printf("\nContact successfully deleted");
        return 0;
    case 8: p=1;
        break;
    default: printf("\nInvalid Input : ");
        break;
}
}
return 0;
```



```
}

int ADD_CONTACT()
{
    int select;
    if(total+1!=MAX_CONTACTS)
    {
        int opt;
        printf("\nEnter number : ");
        scanf("%s",a.number);
        for(i=0;i<total;i++)
        {
            if(strcmp(a.number,all[i].number)==0)
            {
                printf("\nNumber already exist!");
                return 0;
            }
        }

        printf("\nEnter name : ");
        scanf("%s",a.name);
        for(i=0;i<total;i++)
        {
            if(a.name==all[i].name)
            {
                printf("\nA contact under this name already exists");
                return 0;
            }
        }
        printf("\nPress 1.Add more details");
        printf("\nPress 2.Save\n");
        scanf("%d",&opt);
        switch(opt)
        {
            case 1:
                break;
            case 2:
                strcpy(a.address,"__");
                strcpy(a.category,"__");
                strcpy(a.DOB,"__");
                strcpy(a.email,"__");
                all[total]=a;
                total++;
                printf("\nContact added successfully");
        }
    }
}
```

```
    return 0;
    default:
        printf("\nPlease Invalid Input");
    }
    printf("\nEnter address : ");
    scanf("%s",a.address);
    printf("\nEnter Email id : ");
    scanf("%s",a.email);
    printf("\nSelect category : ");
    printf("\n1.Family");
    printf("\n2.Friend");
    printf("\n3.Business");
    printf("\n4.Work");
    printf("\n5.Emergency");
    printf("\n6.Myself");
    printf("\n7.Other\n");
    scanf("%d",&select);
    switch(select)
    {
        case 1:strcpy(a.category,"Family");
        break;
        case 2:strcpy(a.category,"Friend");
        break;
        case 3:strcpy(a.category,"Business");
        break;
        case 4:strcpy(a.category,"Work");
        break;
        case 5:strcpy(a.category,"Emergency");
        break;
        case 6:strcpy(a.category,"Myself");
        break;
        case 7:strcpy(a.category,"Other");
        break;
        default:
            printf("\nInvalid Input");
            break;
    }
    printf("\nEnter DoB(DD-MM-YYYY) : ");
    scanf("%s",a.DOB);
    all[total]=a;
    total++;
    printf("\nContact added successfully");
}
else
```

```
    printf("\nContacts limit exceeded");
    return 0;
}

int SEARCH_CONTACT()
{
    int opt,a,x;
    char temp[20];
    printf("\nEnter name : ");
    scanf("%s",temp);
    for(i=0;i<total;i++)
    {
        if(strcmp(temp,all[i].name)==0)
        {
            a=1;
            break;
        }
    }
    if(a==1)
    {
        printf("\nNumber : %s",all[i].number);
        printf("\nPress 1.Show more details");
        printf("\nPress 2.Exit\n");
        scanf("%d",&x);
        switch(x)
        {
            case 1:
                break;
            case 2: SAVE_CONTACTS();
                break;
            default:
                printf("\nInvalid Input");
        }
        printf("\nAddress : %s",all[i].address);
        printf("\nEmail id : %s",all[i].email);
        printf("\nCategory : %s",all[i].category);
        printf("\nDoB : %s",all[i].DOB);
    }
    else
    {
        return 0;
    }
    printf("\nChoose");
    printf("\n1. Edit");
```

```
printf("\n2. Exit\n");
scanf("%d",&opt);
switch(opt)
{
    case 1:OPTIONS();
        break;
    case 2: SAVE_CONTACTS();
        break;
    default:
        printf("\nInvalid Input");
}
return 0;
}

int ALL_CONTACTS()
{
    int opt;
    if(total>0)
    {
        for(int i=0;i<total;i++)
        {
            printf("\n%d. %s",i+1,all[i].name);
            printf("\t%s",all[i].number);
        }
    }
    else
    {
        printf("\nNo contacts found ");
    }
    printf("\nPress 1.Options");
    printf("\nPress 2.Back");
    printf("\nPress 3.Exit\n");
    scanf("%d",&opt);
    switch(opt)
    {
        case 1: OPT();
            break;
        case 2: START();
            break;
        case 3: SAVE_CONTACTS();
            break;
        default:
            printf("\nPlease Invalid Input");
            break;
    }
}
```

```
    return 0;
}

int OPT()
{
    int x;
    printf("\nPress 1.Delete all contacts");
    printf("\nPress 2.Back");
    printf("\nPress 3.Exit\n");
    scanf("%d",&x);
    switch(x)
    {
        case 1:DELETE_ALL();
        break;
        case 2:ALL_CONTACTS();
        break;
        case 3:
            SAVE_CONTACTS();
    }
    return 0;
}

int DELETE_ALL()
{
    for(i=0;i<total;i++)
    {
        free(all[i].address);
        free(all[i].category);
        free(all[i].DOB);
        free(all[i].email);
        free(all[i].name);
        free(all[i].number);
    }
    printf("\nAll contacts deleted successfully");
    return 0;
}

int DETAILS()
{
    strcpy(all[0].name,"siddu");
    strcpy(all[0].address,"Srinivas Nagar");
    strcpy(all[0].email,"siddarthapeeka2003@gmail.com");
    strcpy(all[0].number,"7569252827");
    strcpy(all[0].category,"Myself");
    strcpy(all[0].DOB,"26-12-2003");
```

```

strcpy(all[1].name,"vignesh");
strcpy(all[1].address,"Gandhi Nagar");
strcpy(all[1].email,"vignesh@gmail.com");
strcpy(all[1].number,"4589249876");
strcpy(all[1].category,"Friend");
strcpy(all[1].DOB,"09-03-2005");
strcpy(all[2].name,"shivani");
strcpy(all[2].address,"Nizamabad");
strcpy(all[2].email,"shivanikamani@gmail.com");
strcpy(all[2].number,"9089234658");
strcpy(all[2].category,"Family");
strcpy(all[2].DOB,"04-09-2004");
strcpy(all[3].name,"tejasai");
strcpy(all[3].address,"Madikonda");
strcpy(all[3].email,"tejasai@gmail.com");
strcpy(all[3].number,"7168451354");
strcpy(all[3].category,"Friend");
strcpy(all[3].DOB,"16-12-2004");
strcpy(all[4].name,"saketh");
strcpy(all[4].address,"Srinivas Nagar");
strcpy(all[4].email,"sakethkumar@gmail.com");
strcpy(all[4].number,"8648348312");
strcpy(all[4].category,"Friend");
strcpy(all[4].DOB,"01-02-2004");
strcpy(all[5].name,"Emergency");
strcpy(all[5].address,"__");
strcpy(all[5].email,"__");
strcpy(all[5].number,"911");
strcpy(all[5].category,"Emergency");
strcpy(all[5].DOB,"__");
return 0;
}

int SAVE_CONTACTS()
{
    remove("C:\\Users\\Sidda\\OneDrive\\Desktop\\contacts.txt");
    ptr=fopen("C:\\Users\\Sidda\\OneDrive\\Desktop\\contacts.txt","w");
    if (ptr == NULL) {
        printf("\nFailed to open file\n");
        return 0;
    }
    for (i = 0; i < total; i++) {
        fprintf(ptr,"\n\nName: %s",all[i].name);
        fprintf(ptr,"\n\tNumber: %s",all[i].number);
    }
}

```

```
fprintf(ptr, "\n\tAddress: %s", all[i].address);  
fprintf(ptr, "\n\tEmail id: %s", all[i].email);  
fprintf(ptr, "\n\tCategory: %s", all[i].category);  
fprintf(ptr, "\n\tDate of Birth: %s", all[i].DOB);  
}  
fclose(ptr);  
printf("\nContacts saved to file\n");  
exit(0);  
return 0;  
}
```

## RESULTS:

```
6.Myself
7.Other
4
```

```
Enter DoB(DD-MM-YYYY) : 01-01-2000
```

```
Contact added successfully
```

```
-----WELCOME TO PHONEBOOK-----
```

```
Choose
```

```
1. Add contact
2. Search contact
3. Show all contacts
4. Delete a contact
5. Exit
2
```

```
Enter name : arjun
```

```
Number : 8106956845
```

```
Press 1.Show more details
```

```
Press 2.Back
```

```
1
```

```
Address : hanamkonda
```

```
Email id : arjun.p@gmail.com
```

```
Category : Work
```

```
DoB : 01-01-2000
```

```
-----WELCOME TO PHONEBOOK-----
```

```
Choose
```

```
1. Add contact
2. Search contact
3. Show all contacts
4. Delete a contact
5. Exit
1
```

```
Enter number : 8106956845
```

```
Enter name : arjun
```

```
Press 1.Add more details
```

```
Press 2.Save
```

```
1
```

```
Enter address : hanamkonda
```

```
Enter Email id : arjun.p@gmail.com
```

```
Select category :
```

```
1.Family
2.Friend
3.Business
4.Work
5.Emergency
```



```
Choose
1. Edit
2. Back
1
```

```
Choose
1. Edit name
2. Edit number
3. Edit address
4. Edit email id
5. Edit category
6. Edit DoB
7. Delete
8. Back
2
```

```
Present number : 8106956845
Enter new number : 9010200584
```

```
Number updated successfully
Choose
1. Edit name
2. Edit number
3. Edit address
4. Edit email id
5. Edit category
6. Edit DoB
```

```
7. Delete
8. Back
8
```

```
-----WELCOME TO PHONEBOOK-----
```

```
Choose
1. Add contact
2. Search contact
3. Show all contacts
4. Delete a contact
5. Exit
3
```

```
1. siddu      7569252827
2. vignesh    4589249876
3. shivani    9089234658
4. tejasai    7168451354
5. saketh     8648348312
6. Emergency  911
7. arjun      9010200584
```

```
Press 1.Options
Press 2.Back
Press 3.Exit
1
```

```
Press 1.Delete all contacts
Press 2.Back
```

Press 3.Exit

2

1. siddu 7569252827
2. vignesh 4589249876
3. shivani 9089234658
4. tejasai 7168451354
5. saketh 8648348312
6. Emergency 911
7. arjun 9010200584

Press 1.Options

Press 2.Back

Press 3.Exit

2

-----WELCOME TO PHONEBOOK-----

Choose

1. Add contact
2. Search contact
3. Show all contacts
4. Delete a contact
5. Exit

4

Enter name : arjun

Press 1.Continue

1. Add contact
2. Search contact
3. Show all contacts
4. Delete a contact
5. Exit

4

Enter name : arjun

Press 1.Continue

Press 2.Cancel

1

Contact successfully deleted

-----WELCOME TO PHONEBOOK-----

Choose

1. Add contact
2. Search contact
3. Show all contacts
4. Delete a contact
5. Exit

5

Contacts saved to file

Thankyou

PS C:\Users\Sidda\OneDrive\Documents\IP> █

Name: siddu  
Number: 7569252827  
Address: Srinivas Nagar  
Email id: siddarthapeeka2003@gmail.com  
Category: Myself  
Date of Birth: 26-12-2003

Name: vignesh  
Number: 4589249876  
Address: Gandhi Nagar  
Email id: vignesh@gmail.com  
Category: Friend  
Date of Birth: 09-03-2005

Name: shivani  
Number: 9089234658  
Address: Nizamabad  
Email id: shivanikamani@gmail.com  
Category: Family  
Date of Birth: 04-09-2004

Name: tejasai  
Number: 7168451354  
Address: Madikonda  
Email id: tejasai@gmail.com  
Category: Friend  
Date of Birth: 16-12-2004

Name: saketh  
Number: 8648348312  
Address: Srinivas Nagar  
Email id: sakethkumar@gmail.com  
Category: Friend  
Date of Birth: 01-02-2004

Name: Emergency  
Number: 911  
Address: \_\_\_\_  
Email id: \_\_\_\_  
Category: Emergency  
Date of Birth: \_\_\_\_